**Alphabet Soup Funding Application Neural Net Model**

**Overview**:

The nonprofit foundation Alphabet soup provides funding for ventures from organizations based on their application. Their business team has provided a CSV with more than 34K organizations that have received funding from Alphabet soup over the years. They have asked for a model to help predict whether applicants will be successful if funded.

**Results**:

The provided csv file contains multiple columns that capture metadata bout each organization. This includes the following:

* EIN and NAME—Identification columns
* APPLICATION\_TYPE—Alphabet Soup application type
* AFFILIATION—Affiliated sector of industry
* CLASSIFICATION—Government organization classification
* USE\_CASE—Use case for funding
* ORGANIZATION—Organization type
* STATUS—Active status
* INCOME\_AMT—Income classification
* SPECIAL\_CONSIDERATIONS—Special considerations for application
* ASK\_AMT—Funding amount requested
* IS\_SUCCESSFUL—Was the money used effectively

Data Processing:

* *What variable(s) are the target(s) for your model?* The target for this model is “Is\_Successful”. We want to determine whether a venture will be successful based on the other given data.
* *What variable(s) are the features for your model?* All other columns in the CSV file were used as features to train the model, except the EIN and Name columns which were removed.
* *What variable(s) should be removed from the input data because they are neither targets nor features?* EIN and Name were removed from the model as they do not help in predicting the successful outcome.

Compiling, Training, and Evaluating the Model

* *How many neurons, layers, and activation functions did you select for your neural network model, and why*?

The model was setup with the below features. In total, there are 5 layers (including the input layer, 3 hidden layers, and output layer), 225 neurons, and 2 different activation functions.

* + Input Layer
  + First Hidden Layer: 128 neurons with ReLU activation function.
  + Second Hidden Layer: 64 neurons with ReLU activation function.
  + Third Hidden Layer: 32 neurons with ReLU activation function.
  + Output Layer: 1 neuron with sigmoid activation function.

3 hidden layers was chosen to try and provide enough depth for the model to learn without becoming too complex that it led to overfitting. The decrease in neurons per layer helps to simplify the features and refine any patterns. ReLU activation function was used in the hidden layers due to the non-linearity of the number of neurons. In contract, Sigmoid was used in the final output layer since the probability is an output of either yes or no (1 or 0).

* *Were you able to achieve the target model performance?* After tweaking all inputs and model adjustments multiple times, the target model performance was unfortunately not achieved.
* *What steps did you take in your attempts to increase model performance?* The first step was to obtain a baseline accuracy from the startup code file of 0.7265. From there, I added increasing number of layers, nodules, and test sizes to try and achieve a goal accuracy of 0.75. Through all the trials performed, the highest achieved was 0.7348.

**Summary**:

The model achieved an accuracy of 0.7348 and Loss of 0.5529. This means the model will predict the success of an organization’s venture around 73.5% of the time. The high loss indicates that further tuning is needed for the model to perform better.

A recommendation for a future model would be to use Keras tuner. This would allow the program to select the best number of hyperparameters instead of manually changing and trying multiple times to get the best result.